University of Burgundy

Master of Science in Computer Vision – 2nd Year

Advanced Image Analysis Module

Project Report on

Performance Evaluation of Various Image Denoising Algorithms

by

Vamshi Kodipaka

Dated: 3rd January 2020

Supervisor:

Prof. Olivier Laligant

UNIVERSITÉ DE BOURGOGNE

CONTENTS

1. INTRODUCTION

Images taken with both digital cameras and conventional film cameras will pick up noise from a variety of sources. Further use of these images will often require that the noise be (partially) removed – for aesthetic purposes as in artistic work or marketing, or for practical purposes such as computer vision. Now I discuss the types of noises shortly. In salt and pepper noise (sparse light and dark disturbances), pixels in the image are very different in color or intensity from their surrounding pixels; the defining characteristic is that the value of a noisy pixel bears no relation to the color of surrounding pixels. Generally this type of noise will only affect a small number of image pixels. When viewed, the image contains dark and white dots, hence the term salt and pepper noise. Typical sources include flecks of dust inside the camera and overheated or faulty CCD elements. In Gaussian noise, each pixel in the image will be changed from its original value by a (usually) small amount. A histogram, a plot of the amount of distortion of a pixel value against the frequency with which it occurs, shows a normal distribution of noise. While other distributions are possible, the Gaussian (normal) distribution is usually a good model, due to the central limit theorem that says that the sum of different noises tends to approach a Gaussian distribution. In either case, the noise at different pixels can be either correlated or uncorrelated; in many cases, noise values at different pixels are modeled as being independent and identically distributed, and hence uncorrelated. So, these paves a way to study of Image De-Noising Techniques in depth.

1.1 What is Image De-Noising?

Image noise is random variation of brightness or color information in images, and is usually an aspect of electronic noise. It can be produced by the sensor and circuitry of a scanner or digital camera. Image noise is an undesirable by-product of image capture that obscures the desired information. **Noise removal** is process of removing or reducing the noise from images. These algorithms reduce or removes visibility of noise by smoothing the entire image leaving areas near contrast boundaries. But these methods can obscure fine, low contrast details[1].

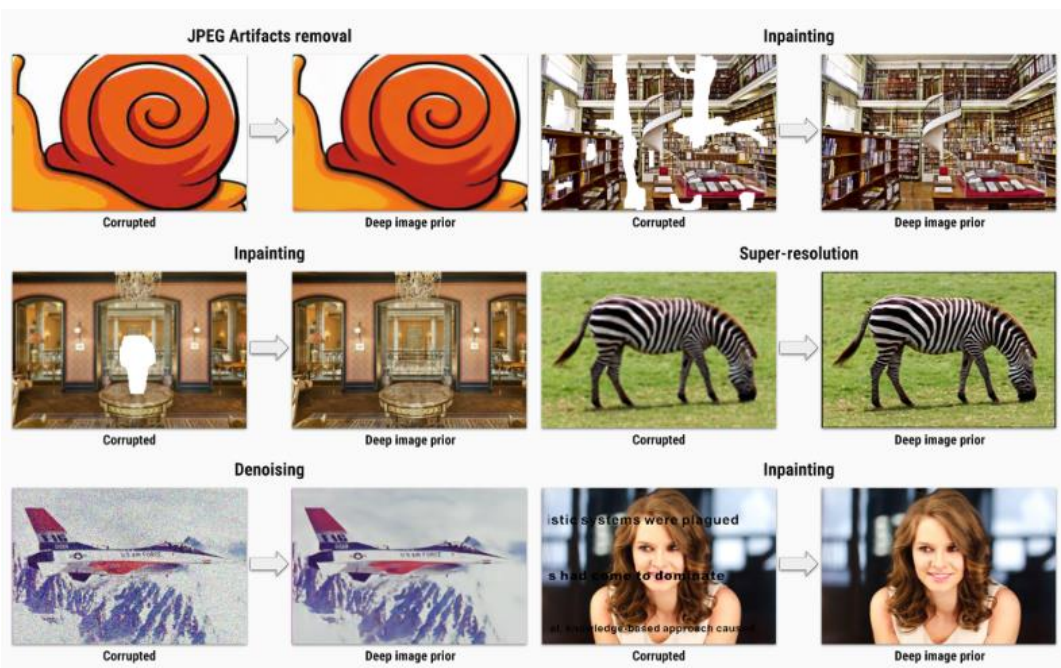


Fig.1 Sample Image Operations

There are many noise reduction algorithms in image processing. In selecting a noise reduction algorithm, one must weigh several factors:

- the available computer power and time available: a digital camera must apply noise reduction in a fraction of a second using a tiny onboard CPU, while a desktop computer has much more power and time
- whether sacrificing some real detail is acceptable if it allows more noise to be removed (how aggressively to decide whether variations in the image are noise or not)
- the characteristics of the noise and the detail in the image, to better make those decisions

*Chroma and luminance noise separation***:** In real-world photographs, the highest spatial-frequency detail consists mostly of variations in brightness ("luminance detail") rather than variations in hue ("chroma detail"). Since any noise reduction algorithm should attempt to remove noise without sacrificing real detail from the scene photographed, one risks a greater loss of detail from luminance noise reduction than chroma noise reduction simply because most scenes have little high frequency chroma detail to begin with. In addition, most people find chroma noise in images more objectionable than luminance noise; the colored blobs are considered "digital-looking" and unnatural, compared to the grainy appearance of luminance noise that some compare to film grain. For these two reasons, most photographic noise reduction algorithms split the image detail into chroma and luminance components and apply more noise reduction to the former. Most dedicated noise-reduction computer software allows the user to control chroma and luminance noise reduction separately.

Linear smoothing filters: One method to remove noise is by convolving the original image with a mask that represents a low-pass filter or smoothing operation. For example, the Gaussian mask comprises elements determined by a Gaussian function. This convolution brings the value of each pixel into closer harmony with the values of its neighbors. In general, a smoothing filter sets each pixel to the average value, or a weighted average, of itself and its nearby neighbors; the Gaussian filter is just one possible set of weights. Smoothing filters tend to blur an image, because pixel intensity values that are significantly higher or lower than the surrounding neighborhood would "smear" across the area. Because of this blurring, linear filters are seldom used in practice for noise reduction; they are, however, often used as the basis for nonlinear noise reduction filters.

Nonlinear filters: A median filter is an example of a non-linear filter and, if properly designed, is very good at preserving image detail. To run a median filter: consider each pixel in the image, then sort the neighboring pixels into order based upon their intensities and finally replace the original value of the pixel with the median value from the list. A median filter is a rank-selection (RS) filter, a particularly harsh member of the family of rank-conditioned rank-selection (RCRS) filters a much milder member of that family, for example one that selects the closest of the neighboring values when a pixel's value is external in its neighborhood, and leaves it unchanged otherwise, is sometimes preferred, especially in photographic applications. Median and other RCRS filters are good at removing salt and pepper noise from an image, and also cause relatively little blurring of edges, and hence are often used in computer vision applications.

Non-local means: This approach for removing noise is based on non-local averaging of all the pixels in an image. In particular, the amount of weighting for a pixel is based on the degree of similarity between a small patch centered on that pixel and the small patch centered on the pixel being de-noised.

Statistical methods: Statistical methods for image denoising exist as well, though they are infrequently used as they are computationally demanding. For Gaussian noise, one can model the pixels in a greyscale image as auto-normally distributed, where each pixel's "true" greyscale value is normally distributed with mean equal to the average greyscale value of its neighboring pixels and a given variance. One method of denoising that uses the auto-normal model uses the image data as a Bayesian prior and the auto-normal density as a likelihood function, with the resulting posterior distribution offering a mean or mode as a denoised image

Block-matching algorithms: A block-matching algorithm can be applied to group similar image fragments into overlapping macroblocks of identical size, stacks of similar macroblocks are then filtered together in the transform domain and each image fragment is finally restored to its original location using a weighted average of the overlapping pixels.

Anisotropic diffusion: Another method for removing noise is to evolve the image under a smoothing partial differential equation similar to the heat equation, which is called anisotropic diffusion. With a spatially constant diffusion coefficient, this is equivalent to the heat equation or linear Gaussian filtering, but with a diffusion coefficient designed to detect edges, noise can be removed without blurring edges of the image.

Random field: Shrinkage fields is a random field-based machine learning technique that brings performance comparable to that of Block-matching and 3D filtering yet requires much lower computational overhead (such that it could be performed directly within embedded systems)

Deep learning based: Various deep learning approaches have been proposed to solve noise reduction and such image restoration tasks. Deep Image Prior is one such technique which makes use of convolutional neural network and is distinct in that it requires no prior training data

Software programs: Most general purpose image and photo editing software will have one or more noise reduction functions (median, blur, despeckle, etc.). Special purpose noise reduction software programs include Neat Image, Noiseless, Noiseware, Noise Ninja, G'MIC (through the -*denoise* command), and pnmnlfilt (nonlinear filter) found in the open source Netpbm tools. General purpose image and photo editing software including noise reduction functions include Adobe Photoshop, GIMP, Photo Impact, Paint Shop Pro, Helicon Filter, and Dark table[1]

1.2 Why Image Transforms?

Mathematical transformations are applied to image to obtain further information from that image that is not readily available in the raw image. There are a number of transformations that can be applied, among which the Fourier transforms are probably by far the most popular.

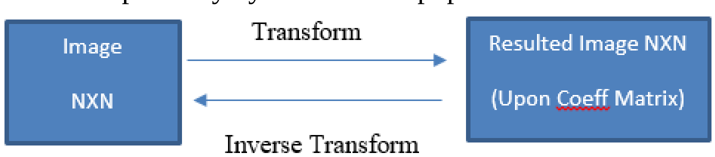


Fig.2 Image Transformation

From Fig.2, where the input is an image and after the image is transformed, we get another image. So if the size of the input image is NxN, say it is having N number of rows and N number of columns. The transformed image is also of same size, that of size NxN. And given this transformed image, if we perform the inverse transformation, we get back the original image. That is, image of size NxN. Now if given an image by applying transformation, we are transforming back to another image of same size and doing the inverse transformation operation we get back the original image, then the question naturally comes that what is the use of this transformation[2]. And here you find that after transformation the second image of same size NxN that we get that is called the transformed coefficient matrix. Now we will find and we will also see on in our subsequent sections that this kind of transformation has got a number of very, very important applications. One of the application is for preprocessing, in case of image preprocessing of the images.

From Joseph Fourier to Jean Morlet and after ... a story almost French ancestors worked decades. The Fourier transform gives the frequency components (spectral) exists in the signal. But, when the **time-localization** of spectral components is needed, a transform giving the TIME-FREQUENCY representation of signal is needed. This is in short, if we take the Fourier transform over the whole time axis, we cannot tell at what instant a particular frequency rises. The wavelet transform is a transform which gives this sort of information. There are other transforms which give this information too, like Short-time Fourier transform (STFT) uses a sliding window to find spectrogram, which gives the information of both time and frequency. But still another problem exists: The length of window limits the resolution in frequency. Wavelet transform seems to be a solution to the problem above.

2. Discrete Wavelet Transform based Denoising

Wavelet Transform has the main aim of an image denoising algorithm is to achieve both noise reduction and feature preservation. In this context, wavelet-based methods are of particular interest. In the wavelet domain, the noise is uniformly spread throughout coefficients while most of the image information is concentrated in a few large ones. Therefore, the first wavelet-based denoising methods were based on thresholding of detail sub-bands coefficients. However, most of the wavelet thresholding methods suffer from the drawback that the chosen threshold may not match the specific distribution of signal and noise components at different scales and orientations.

To address these disadvantages, non-linear estimators based on Bayesian theory have been developed. In the Bayesian framework, it has been recognized that a successful denoising algorithm can achieve both noise reduction and feature preservation if it employs an accurate statistical description of the signal and noise components.

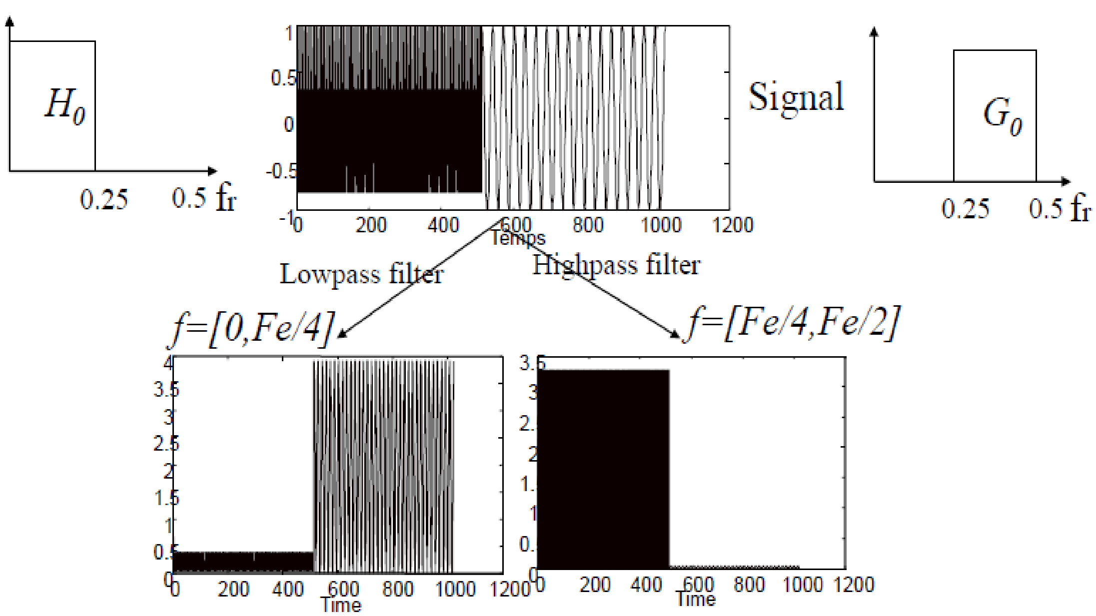


Fig.3 Self Explanatory example for Discrete Wavelet Decomposition

Discrete Wavelet Transform

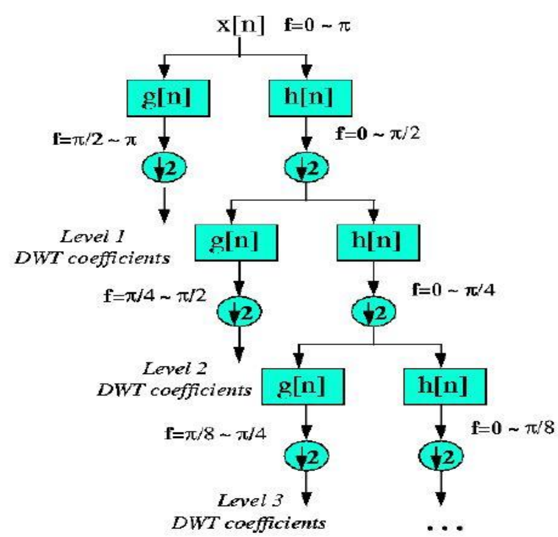
Discrete wavelet transform (DWT) provides sufficient information both for analysis and synthesis of the original signal, with a significant reduction in the computation time. The DWT is considerably easier to implement when compared to the Continuous Wavelet Transform (CWT).

Aim: Study of the signal frequency behavior during the time.
How: By using some filtering operations.

The main idea is the same as it is in the Continuous Wavelet Transform. A time-scale representation of a digital signal is obtained using digital filtering techniques. The CWT is a **correlation between a wavelet at different scales and the signal with the scale (or the frequency) being used as a measure of similarity**. The continuous wavelet transform is computed by changing the scale of the analysis window, shifting the window in time, multiplying by the signal, and integrating over all times. In the discrete case, filters of different cutoff frequencies are used to analyze the signal at different scales. The signal is passed through a series of high pass filters to analyze the high frequencies, and it is passed through a series of low pass filters to analyze the low frequencies. The **resolution of the signal**, which is a measure of the amount of detail information in the signal, **is changed by the filtering operations**, and the **scale is changed by up sampling and down sampling operations**. Down sampling a signal corresponds to reducing the sampling rate, or removing some of the samples of the signal. Up sampling a signal corresponds to increasing the sampling rate of a signal by adding new samples to the signal[5].

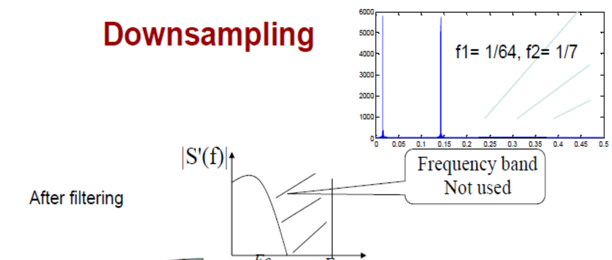
2.1 Wavelet Analysis (Decomposition)

For example, the procedure starts with passing this signal through a **half band digital low pass filter** with impulse response h[n]. A half band low pass filter removes all frequencies that are above half of the highest frequency in the signal. For example, if a signal has a maximum of 500 Hz component, then half band low pass filtering removes all the frequencies above 250 Hz.



After passing the signal through a half band low pass filter, half of the samples can be eliminated according to the Nyquist's rule, since the signal now has a highest frequency of p/2 radians instead of p radians (if p is original highest frequency of signal). Simply discarding every other sample will **down sample** the signal by two, and the signal will then have half the number of points. **The scale of the signal is now doubled**. Here, the low pass filtering removes the high frequency information, but leaves the scale unchanged. Only the down sampling process changes the scale. See Fig above.

The DWT analyzes the signal at different frequency bands with different resolutions by decomposing the signal into a **coarse approximation** and **detail information**. DWT employs two sets of functions, called **scaling functions** and **wavelet functions**, which are associated with **low pass** and **high pass** filters, respectively. The decomposition of the signal into different frequency bands is simply obtained by successive high pass and low pass filtering of the time domain signal. The original signal x[n] is first passed through a half band high pass filter g[n] and a low pass filter h[n]. After the filtering, half of the samples can be eliminated. This constitutes one level of decomposition. This decomposition can be continued to multi-level using the coarse approximation as input to next level[6].



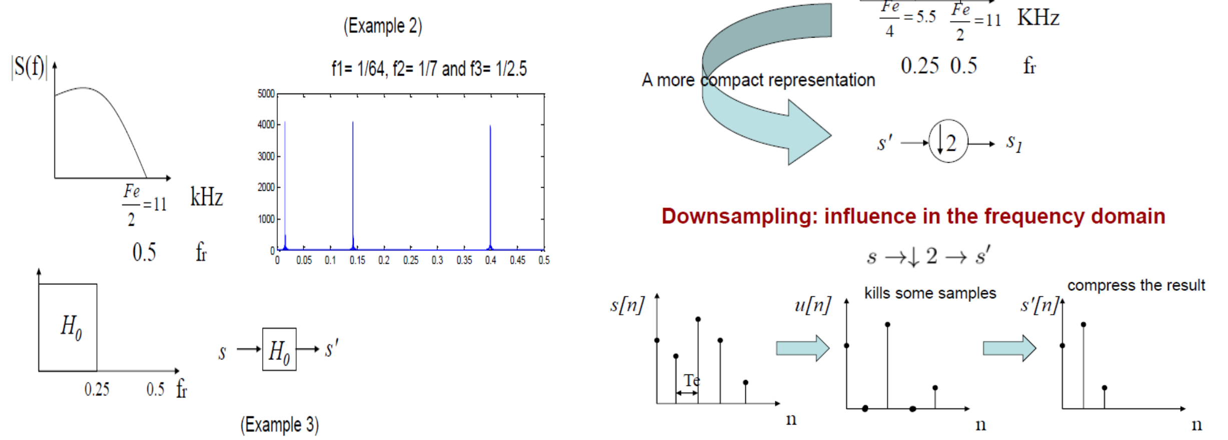


Fig. 4 Wavelet Decomposition of 1D signal & Downsampling

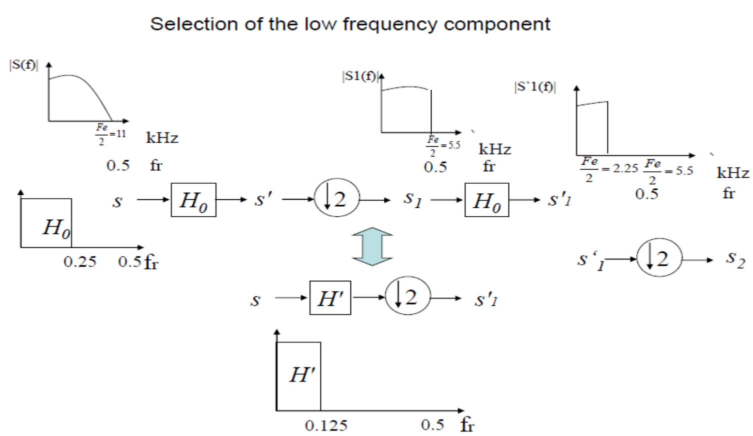
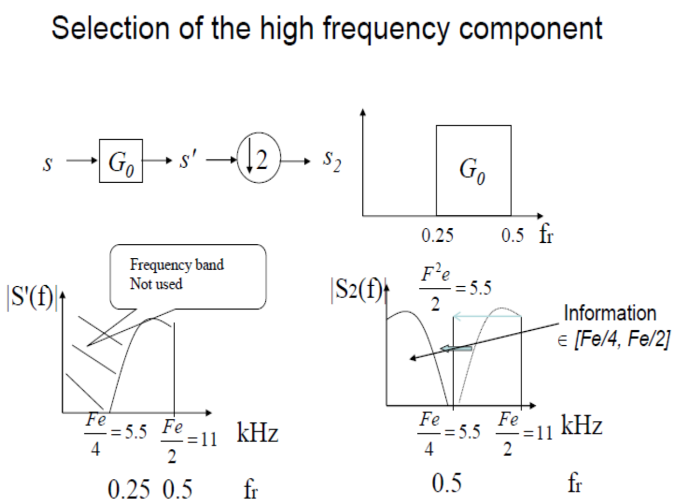


Fig.5 Low-Pass-Filter Selection for Downsampling

2.2 Wavelet Synthesis (Reconstruction)

The reconstruction of original signal from the wavelet coefficients is the reverse process of the decomposition. As we get the wavelet coefficients, we need to up sample the coefficients which halves the scale of the signal. After the signal is up sampled we apply low pass and high pass filters to the up sampled signal to get the perfect reconstructed signal.



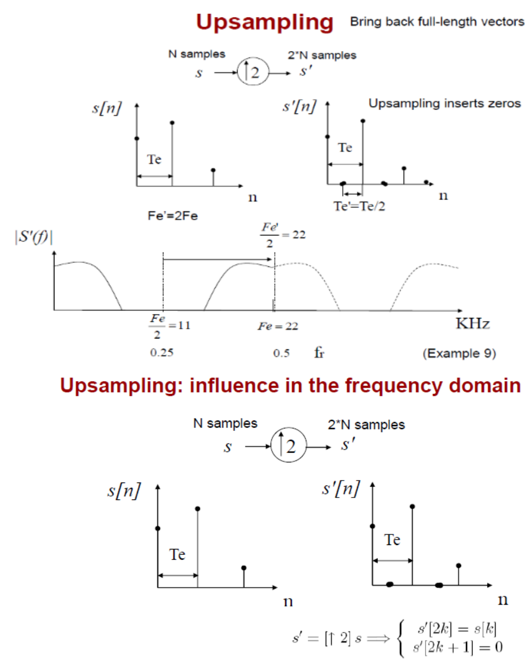


Fig.6 High-Pass-Filter Selection for Upsampling

2.3 2D Application of Wavelets:

2D wavelet analysis (decompostion):

For images, an algorithm similar to the one-dimensional case is possible for two-dimensional wavelets. This kind of two-dimensional DWT leads to a decomposition of approximation coefficients at level *j* in four components: the approximation at level *j*+1, and the details in three orientations (horizontal, vertical, and diagonal). We apply low pass filter on rows of an image and then down sample the columns by half and similarly we apply high pass filter and down sample the columns by half. Further, we apply low pass filter on columns of result obtained by down sampling columns, and then we down sample the rows. Similarly high pass filter is also applied on the result after down sampling the columns, and then we down sample the rows.

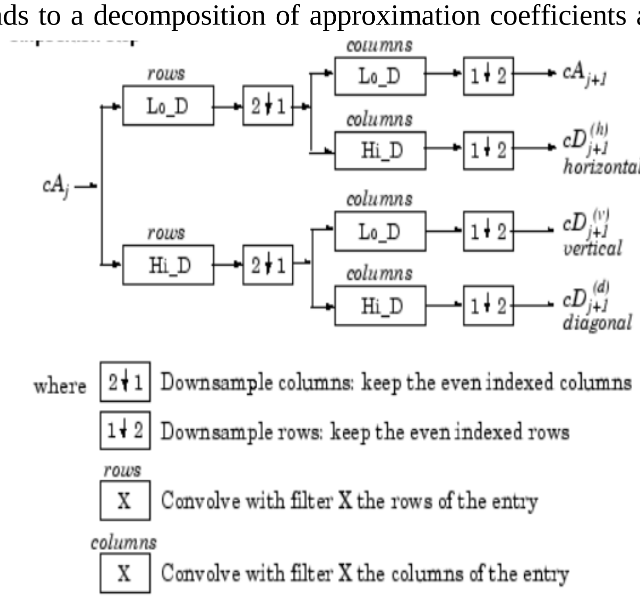


Fig. 7 Schematic diagram of 2D DWT decomposition

So here the 2D signal is divided into four bands LL, HL, LH, HH , where HL band indicated the variation along the horizontal while the LH band shows the vertical variation.

LL	HL
LH	HH

Fig. 8 Four bands in 2D signal after decomposition

I have used **Daubechies D4 filter as low pass filter. the high pass filter is computed by,**

$$\boldsymbol{H(z) = G(-z)}$$

This is achieved by making the even terms of D4 filter as negative.

The code for decomposition is developed and it is available as a function:

[C, S, wc] = discreteWavletTrans(I, J, lpfCoeff) (details are given in function)

2.4 2D wavelet synthesis (reconstruction):

It is as reverse process to analysis process. We take the four band coefficients of decomposition and up sample the rows of four bands and apply low pass filter on columns of LL and HL bands and high pass filter on columns of LH and HH bands. We then add the results as shown in fig. 4, and then up sample the columns and apply low pass and high pass filters along rows and add the result to get the final reconstructed result[6].

The code for reconstruction is developed and it is available as a function:

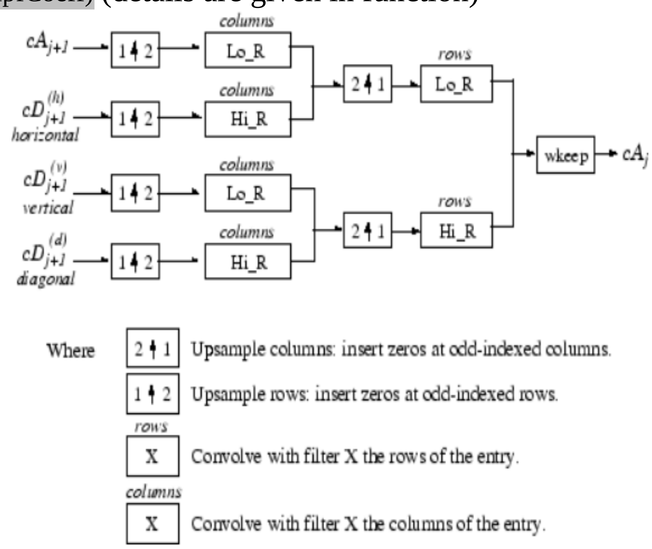


Fig. 9 Schematic diagram of 2D DWT reconstruction

imageReconst = InvdiscreteWavletTrans(C, S, J, lpfCoeff) (details are given in function)

The multi-level wavelet transform is also similar to single level, but we consider the approximation as input to the next level in decomposition.

Note: The low frequency band is splitted into small segments in order to separate all the components of the signal. The high frequency band is splitted into large segments

Properties of Wavelet decomposition:

1. The wavelet transforms of real-world images tend to be sparse. A wavelet coefficient is large only if edges are present within the support of the wavelet.
2. The noise remains the same, the small coefficient are more likely due to noise.
3. Wavelet provides an appropriate basis for separating noisy signal from the image signal.

2.4 Results of wavelet transform on a Sample Image

I tried to input, synthetic input of random values as a 2D input. I decomposed the input using the developed function and reconstructed the wavelet coefficients[7] and calculated the error between the original input an reconstructed result, which **results in 10^--5.** The results of wavelet decomposition and reconstruction of Cameraman image is figs.10-13.

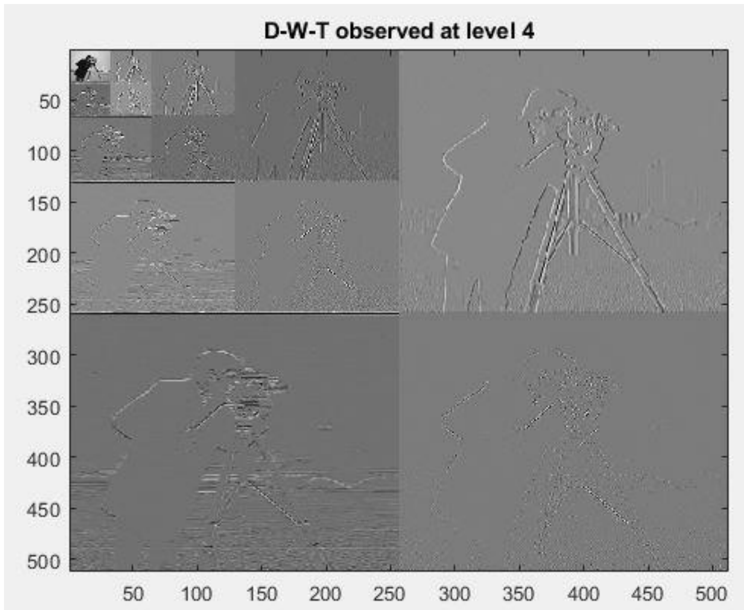


Fig. 10 Discrete wavelet decomposition at level 4 of Lena image

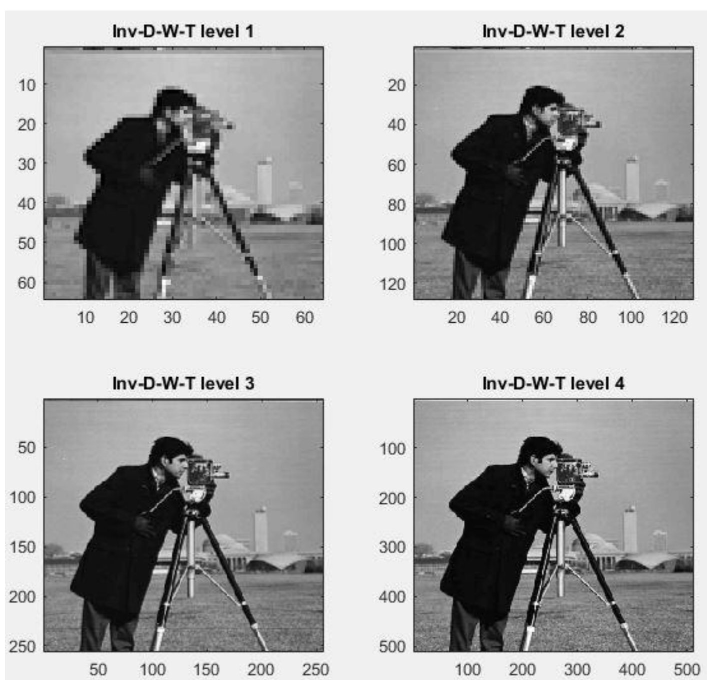


Fig. 11 DW reconstruction of Cameraman at 4 levels

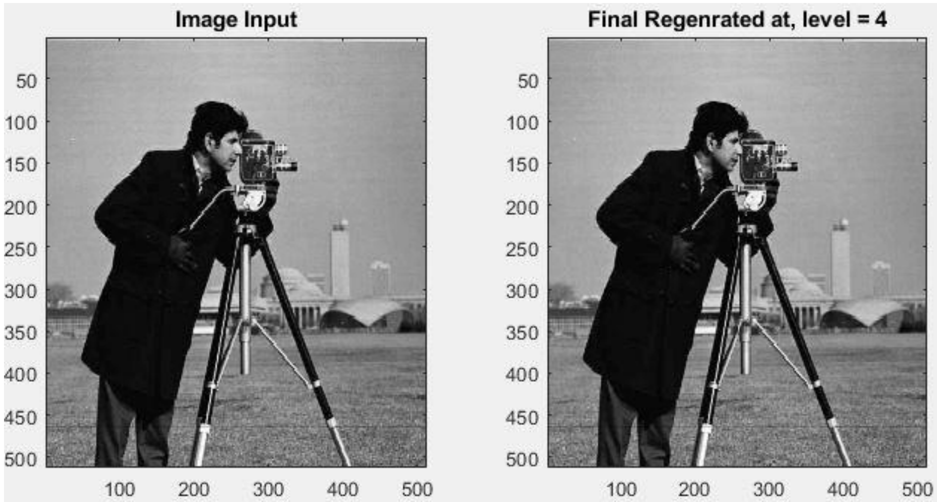


Fig. 12 Original and reconstructed image using DWT at level 4

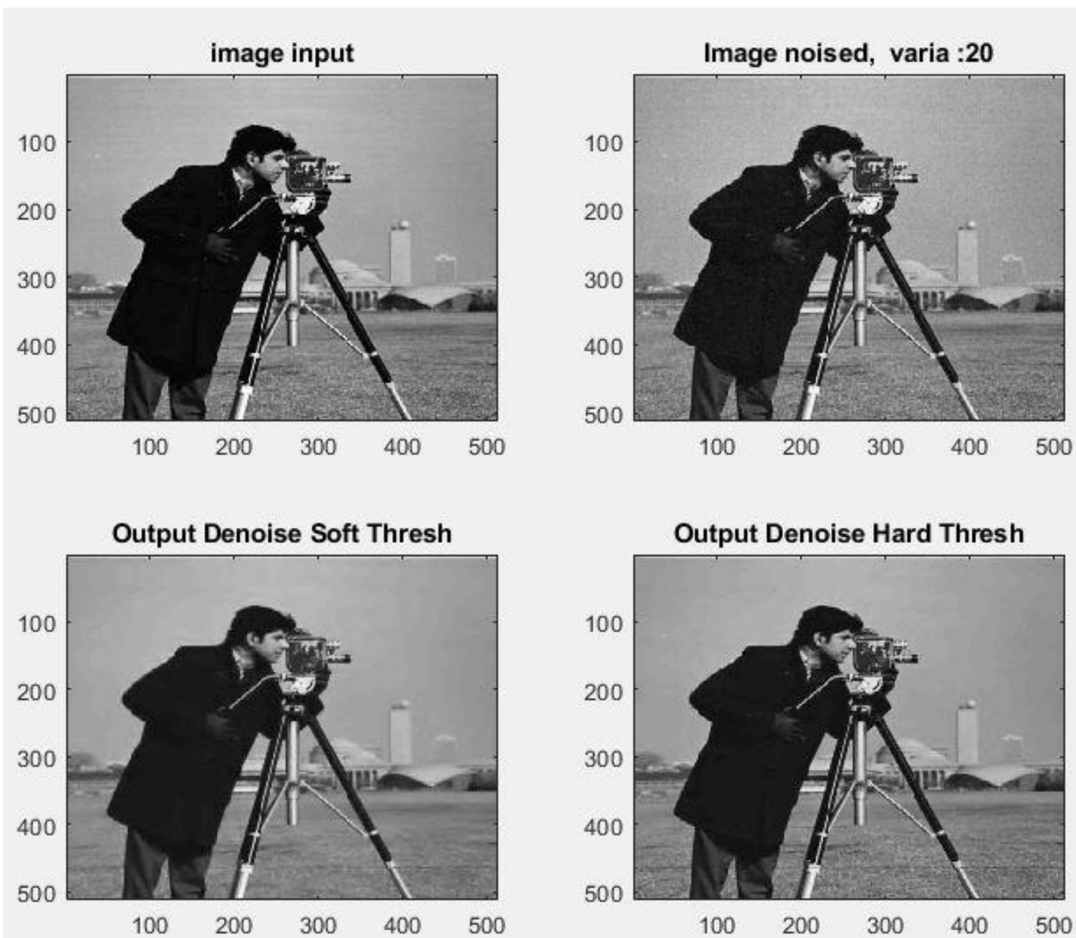


Fig. 13 Variance (Var:20) and Means (const.) handling through Soft and Hard Thresholds

Soft thresholding, shrinks coefficients above the threshold in absolute value. While at first sight hard thresholding may seem to be natural, the continuity of soft thresholding has some advantages. It makes algorithms mathematically more tractable. Sometimes, pure noise coefficients may pass the hard threshold and appear as annoying "blips" in the output. Soft thresholding shrinks these false structures.

Soft thresholding provides smoother results in comparison with the hard thresholding. Hard threshold, however, provides better edge preservation in comparison with the soft one. I varied the variance from 0 to 20 and plotted the mean square error between the original and denoised images. The plot is as shown in fig. 13 for soft and hard thresholding.

The principal work on denoising is based on thresholding the DWT of the signal. The method relies on the fact that noise commonly manifests itself as fine-grained structure in the signal, and WT provides a scale-based decomposition[3]. Thus, most of the noise tends to be represented by the wavelet coefficients at finer scales. Discarding these coefficients would result in a natural filtering out of noise on the basis of scale. Because the coefficients at such scale also tend to be the primary carriers of edge information, the method of Donoho, thresholds the wavelet coefficients to zero if their values are below a threshold. These coefficients are mostly those corresponding to the noise. The edge related coefficients of the signal on the other hand, are usually above the threshold. Several approaches have been suggested for setting the threshold for each band of the wavelet decomposition. A common approach is to compute the sample variance of the coefficients in a band and set the threshold to some multiple of the deviation, which is given by the universal thresholding method **'sqtwolog'** method [3].

$$\boldsymbol{\lambda} = \boldsymbol{\sigma}\sqrt{\mathbf{2}\ \mathbf{log}\,\boldsymbol{n}}, where\ n\ is\ no.of\ data\ points$$

$$\lambda = 3\sigma$$

$$where, \sigma = \frac{median(|w|)}{0.6745}$$

$$where, w\ are\ wavelet\ coefficients$$

There are two types of thresholding [2]:

1. Hard thresholding

$$T^{hard}(d_{j,k}) = \begin{cases} d_{j,k} if\ |d_{j,k}| \geq \lambda \\ 0\ if\ |d_{j,k}| < \lambda \end{cases}$$

2. Soft Thresholding

$$T^{soft}(d_{j,k}) = \begin{cases} sign(d_{j,k})(|d_{j,k}| - \lambda) if\ |d_{j,k}| \geq \lambda \\ 0\ if\ |d_{j,k}| < \lambda \end{cases}$$

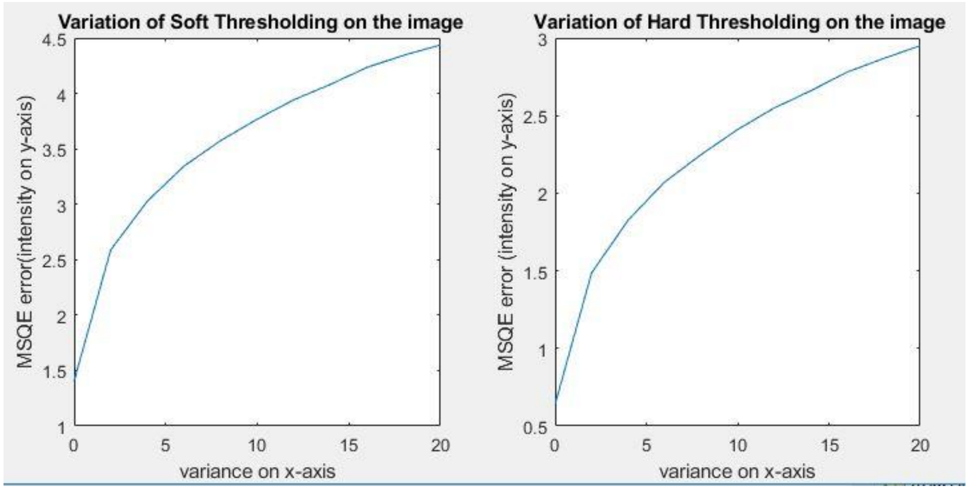


Fig. 14 Mean square error vs variance for soft and hard threshold

TABLE 1 TIME TAKEN BY CODE

Decomposition & Reconstruction	Time (in sec)
Level = 1	2.4
Level = 4	5.6

3. Tikonov's based De-noising
(GAUSS-SEIDEL AND CONJUGATE GRADIENT METHOD)

Tikhonov model[8] is to seek a function u implementing Lagrange Multipliers. Thus, we have the optimization problem:

$$\min_{u \in S} \mathcal{F} = \int_{\Omega} |\nabla u|^2 + \frac{\lambda}{2}(u - u_0)^2 \, dx.$$

Here u carries the image internsiites

We have the G^ateaux derivative, we can apply Theorem 1 and solve for u, which gives us the following Boundary Value Problem.

$$\begin{cases} -2\Delta u + \lambda(u - u_0) = 0 \\ \frac{\partial u}{\partial \vec{n}} = 0 \end{cases}$$

We can now consider our elliptic di_erential equation as a time-dependent parabolic di_erential equation. For convenience, we rewrite the given noisy signal u0 as u0.

$$\begin{cases} \frac{\partial u}{\partial t} = 2\Delta u - \lambda(u - u^0) \\ \frac{\partial u}{\partial \vec{n}} = 0 \\ u(x, t = 0) = u^0 \end{cases}$$

From PDE and boundary conduitions we find solutions, Implementing this scheme essentially gives us Au = b.

Normally, the discrete Laplacian with Neumann conditions is singular. However, with nonzero _, the matrix becomes nonsingular. Direct solution techniques can be costly for very large matrices, even if they are sparse. To deal with this, we will implement Gauss-Seidel and Conjugate Gradient method[8].

3.1 Algorithm :

1. Input the Image. Laod original image and noise image
2. Fix nodes, Max.no.of iterations and convergence tolerence
3. Neumann condition's algebraic constraints
4. Creating 2D Elliptical Operator
5. Then finding kron for 2D dimensional
6. Directing to solving the eqn for comparison PDE
7. Gauss-Seidel usage for inversion of lower-triangular matrix over image
8. Applying Conjugate Gradient now

It is iterative and convergence time and to achieve the computations are high.

3.2 Results of Tikonov's Denoising on a Sample Image

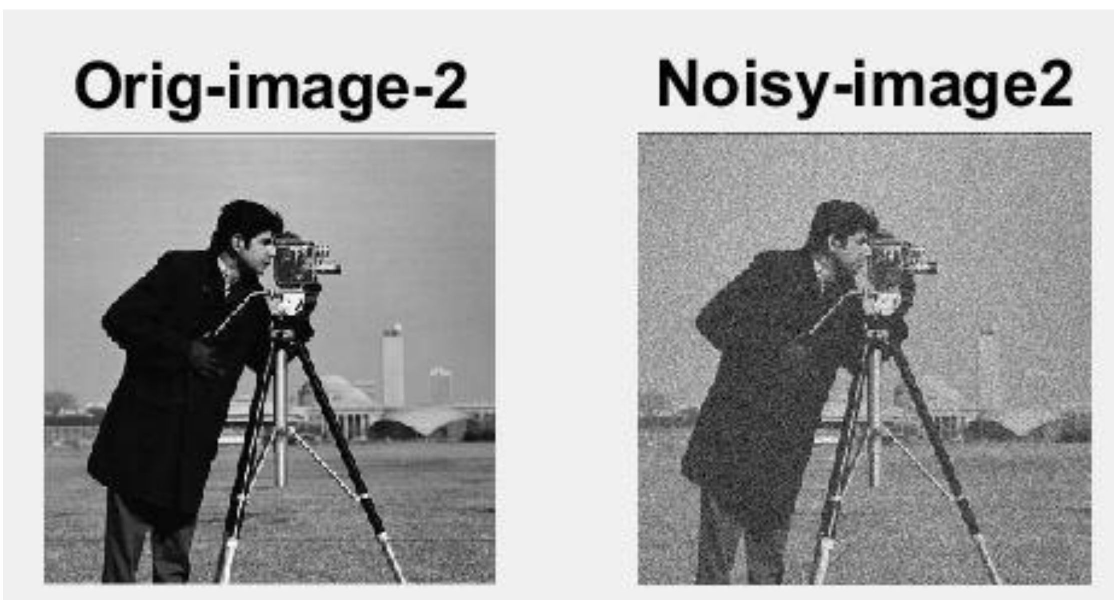


Fig.15 Image inputs: original and Noisy images

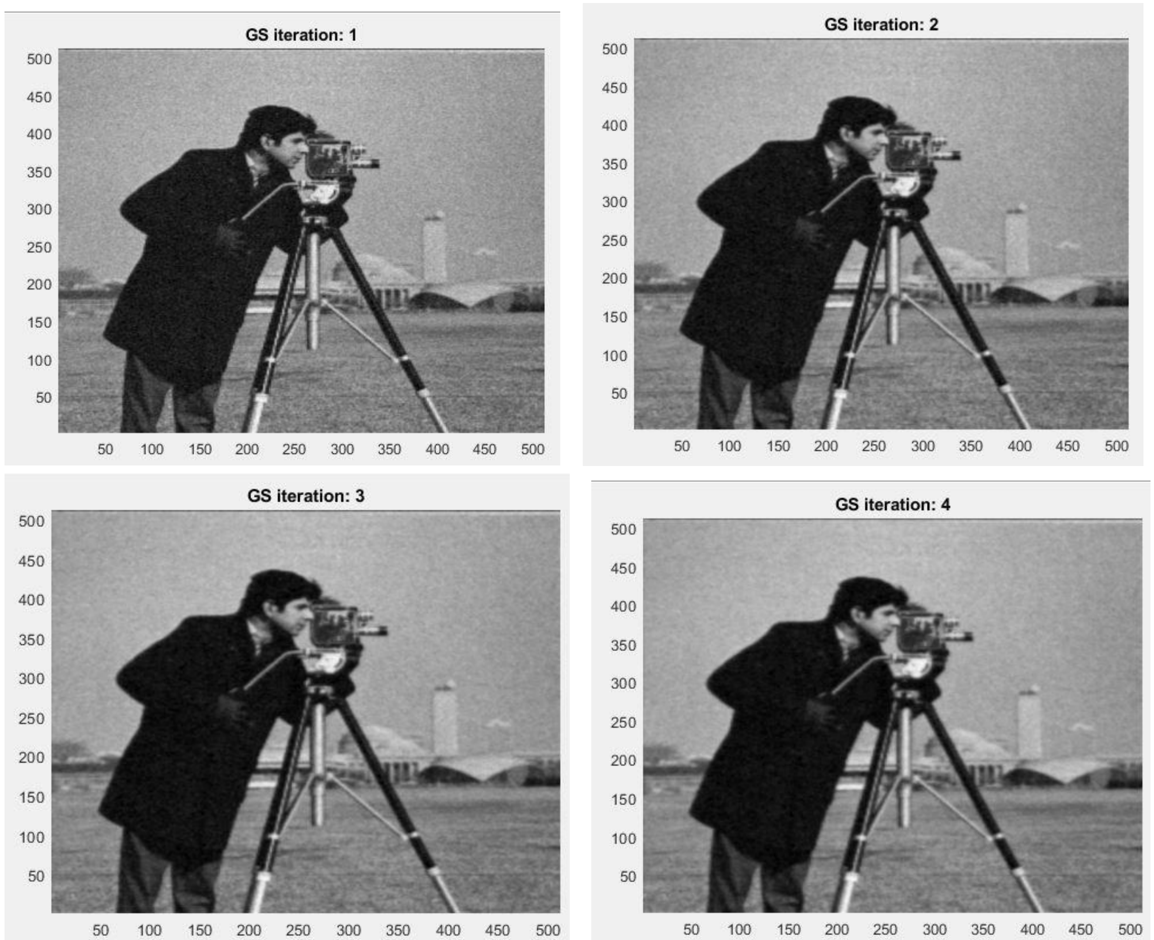


Fig.16 Iterations for GSCD – Tikonov's denoising - Outputs

4. Wiener Filter based De-noising

The Wiener filter is a filter used to produce an estimate of a desired or target random process by linear time-invariant (LTI) filtering of an observed noisy process, assuming known stationary signal and noise spectra, and additive noise. The Wiener filter minimizes the mean square error between the estimated random process and the desired process. More of Wiener filter in 2D is a DFT version on the scale of image[9].

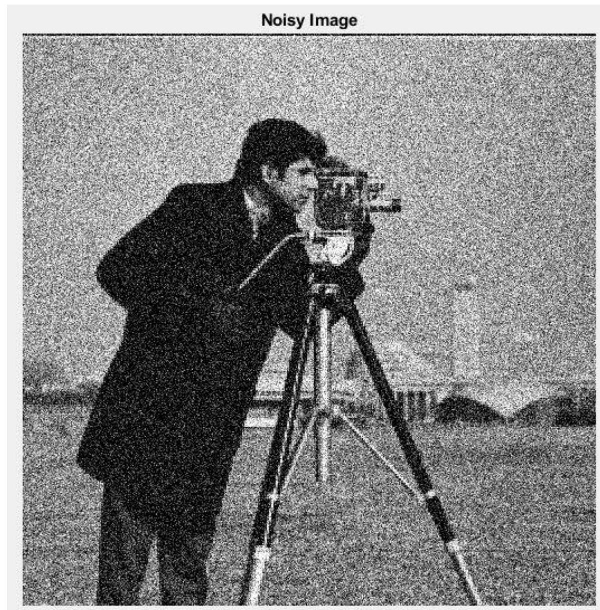
4.1 Algorithm:

1. Read the input image
2. Find the input image Power Spectral Density (by FFT)
3. Add noise and find PSD for the noisy image (by IFT)
4. Applying Gaussian smoothing on noisy image
5. Multiplying by wiener filter in the frequency domain

Wienerfil= (smootedInvPSD(x,y)/(smootedInvPSD(x,y)+phi));

6. Inverse DFT to obtain the wiener filtered image
7. Plot the outputs

4.2 Results of Tikonov's Denoising on a Sample Image



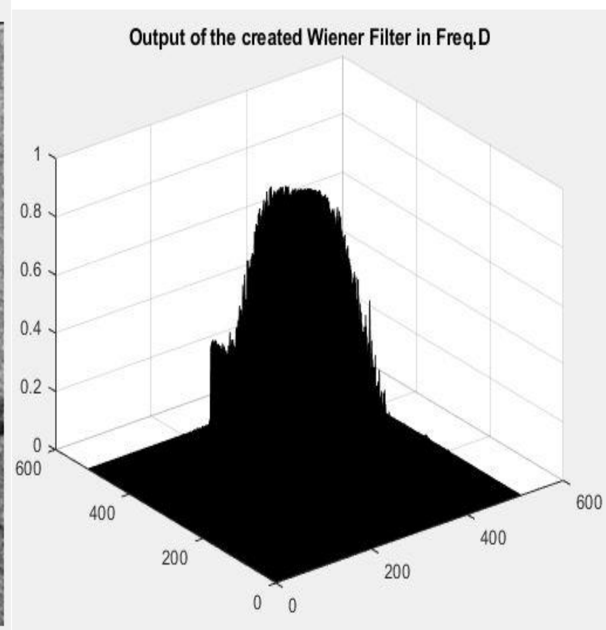


Fig.16 Outputs for the Wiener Filter based denoising

5. Discussions, Applications & Interests

The data sets of many scientific experiments are corrupted with noise, either because of the data acquisition process, or because of environmental effects. A first pre-processing step in analyzing such datasets is denoising, that is, estimating the unknown signal of interest from the available noisy data[4]. There are several different approaches to denoise signals and images. Generally smoothing removes high frequency and retains low frequency (with blurring). De-blurring increases the sharpness signal features by boosting the high frequencies, whereas denoising tries to remove whatever noise is present regardless of the spectral content of a noisy signal.

1. Wavelet transforms enable us to represent signals with a high degree of sparsity. This is the principle behind a non-linear wavelet based signal estimation technique known as wavelet denoising. Wavelet denoising attempts to remove the noise present in the signal while preserving the signal characteristics, regardless of its frequency content.

2. Discrete Wavelet transform is computationally competence which makes it very interesting. It has many applications like denoising, compression, security etc. In image denoising the soft threshold gives smooth result but it does not preserve edges whereas hard threshold preserve edges but it does not remove noise as perfectly as soft thresholding.

3. Gauss-Siedel conjugate gradient or Tikonov approach is iteration based and computational calculations are more. Hence also takes too much time in providing the results.

4. Wiener based De-noising is FFT based and manually chosen for the power spectrum and smoothing filter influence the degree of noise for the output.

Performances:

Algorithm	Execution time	Computation	Pixel level accuracy
Wavelets	5.6 sec	Fixed or minimal	92.5%
Tikonov approach	125mins	Iterative process	72.1%
Wiener	3.75sec	Simple FFT	86.3%

6. Conclusion

Thus, I implemented three algorithms and tested their performance.

7. References

[1] Wikipedia – Image Noise Reduction: https://en.wikipedia.org/wiki/Noise_reduction

[2] IIT NPTEL-Image Denoising Lectures:: https://drive.google.com/file/d/12-GkYGa84GXCBzm7lpYrUVCfQAKnhjOh/view

[3] Jeena Roy, Salice Peter and Neetha John, "Denoising using Soft Thresholding", in *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering* Vol. 2, Issue 3, March 2013.

[4] Wavelet based image demoising: https://pdfs.semanticscholar.org/1e5c/ec5f16940e8e7d8bdb10779ef36cbb2ab4db.pdf

[5] Problem Statement for Wavelet Denoising: https://github.com/gopi231091/Wavelet-Transform-and-Application-to-Image-Denoising/

[6] Wavelet Denoising: https://arxiv.org/ftp/arxiv/papers/1608/1608.00265.pdf

[7] Review on Image Denoising using Wavelet Transforms:

https://www.academia.edu/6801849/A_Review_on_Image_Denoising_using_Wavelet_Transform

[8] "A Variational Approach to Signal Denoising"

https://github.com/gjlper/Tikhonov-Denoising/blob/master/Project1_Report.pdf

[9] Wiener Filtering: https://en.wikipedia.org/wiki/Wiener_filter

8. Appendices

For Appendix and code explanations, see the attached 'appendix.pdf'